

Segmentasi Citra Ikan Tuna dengan Mahalanobis Histogram Thresholding dan Mahalanobis Fuzzy C-Means

Andi Baso Kaswar¹, Agus Zainal Arifin², Arya Yudhi Wijaya³

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya 60111, Jawa Timur
Email: ¹kaswar.andi.baso14@mhs.if.its.ac.id, ²agusza@cs.its.ac.id, ³arya@cs.its.ac.id

Masuk: 19 Januari 2016; Direvisi: 30 Januari 2016; Diterima: 2 Februari 2016

Abstract. Fuzzy C-Means segmentation algorithm based on Mahalanobis distance can be utilized to segment tuna fish image. However, initialization of pixels membership value and clusters centroid randomly cause the segmentation process become inefficient in terms of iteration and time of computation. This paper proposes a new method for tuna fish image segmentation by Mahalanobis Histogram Thresholding (M-HT) and Mahalanobis Fuzzy C-Means (MFCM). The proposed method consists of three main phases, namely: centroid initialization, pixel clustering and accuracy improvement. The experiment carried out obtained average of iteration amount is as many as 66 iteration with average of segmentation time as many as 162.03 second. While the average of Accuracy is 98.54%, average of Missclassification Error is 1.46%. The result shows that the proposed method can improve the efficiency of segmentation method in terms of number of iterations and time of segmentation. Besides that, the proposed method can give more accurate segmentation result compared with the conventional method.

Keywords: Tuna Fish Image, Segmentation, Fuzzy Clustering, Histogram Thresholding, Mahalanobis Distance.

Abstrak. Algoritma segmentasi Fuzzy C-Means berbasis jarak Mahalanobis dapat digunakan untuk mensegmentasi ikan tuna. Namun, inisialisasi derajat keanggotaan piksel dan centroid klaster secara random mengakibatkan proses segmentasi menjadi tidak efisien dalam hal iterasi dan waktu komputasi. Penelitian ini mengusulkan metode baru untuk segmentasi citra ikan tuna dengan Mahalanobis Histogram Thresholding (M-HT) dan Mahalanobis Fuzzy C-Means (MFCM). Metode ini terdiri atas tiga tahap utama, yaitu: inisialisasi centroid, pengklasteran piksel dan peningkatan akurasi. Berdasarkan hasil ekseprimen, diperoleh rata-rata jumlah iterasi sebanyak 66 iterasi dengan rata-rata waktu segmentasi 162,03 detik. Rata-rata Akurasi 98,54% dengan rata-rata tingkat Missclassification Error 1,46%. Hasil yang diperoleh menunjukkan bahwa metode yang diusulkan dapat meningkatkan efisiensi metode segmentasi dalam hal jumlah iterasi dan waktu segmentasi. Selain itu, metode yang diusulkan dapat memberikan hasil segmentasi yang lebih akurat dibandingkan dengan metode konvensional.

Kata Kunci: Citra Ikan Tuna, Segmentasi, Fuzzy Clustering, Histogram Thresholding, Jarak Mahalanobis.

1. Pendahuluan

Ikan tuna adalah salah satu sumber daya alam Indonesia yang cukup berlimpah. Hal ini dibuktikan dengan banyaknya perusahaan yang memproduksi hasil olahan ikan tuna. Salah satu tahapan penting pada proses produksi adalah tahap klasifikasi jenis ikan tuna. Sebelumnya telah diusulkan sistem klasifikasi menggunakan *Fuzzy Decision Tree* untuk melakukan klasifikasi ikan tuna secara otomatis (Puspita, dkk., 2015). Sistem klasifikasi ini mampu memberikan hasil klasifikasi yang cukup akurat. Namun, metode yang digunakan pada tahapan segmentasi berupa deteksi tepi sobel dan operasi morfologi sensitif terhadap *noise*, cahaya yang tidak merata serta varian warna yang tinggi pada ikan tuna. Hal ini mengakibatkan terjadinya kesulitan untuk mendapatkan bagian objek ikan tuna secara utuh. Oleh karena itu, perlu dilakukan perbaikan pada tahap segmentasi agar dapat diperoleh hasil ekstraksi fitur yang lebih optimal.

Terdapat beberapa metode yang bisa dilakukan untuk mengimplementasikan segmentasi. Beberapa metode yang ada dapat dikelompokkan ke dalam tiga kategori berdasarkan dasar matematikanya, yaitu: *threshold-based* (Arifin & Asano, 2006; Ghosh, dkk., 2010; Jati, dkk., 2014), *cluster-based* (Yao, dkk., 2013; Zhao, dkk., 2015a) dan *statistic-based* (Liu, dkk., 2013). Dari ketiga dasar metode tersebut, segmentasi berbasis kluster merupakan metode yang paling banyak digunakan. Salah satu metode segmentasi berbasis kluster adalah *Fuzzy C-Means* (FCM) (Zhao, dkk., 2015b). FCM konvensional menggunakan jarak *Euclidean* untuk membedakan data dan prototipe kluster (Bezdek, dkk., 1984). Metode ini dapat mengelompokkan data dengan akurat namun sensitif terhadap *noise*. Oleh karena itu, beberapa peneliti mengusulkan metode baru dalam rangka meningkatkan performa FCM. Berdasarkan sudut pandang ketetanggaan piksel, diusulkan FCM dengan *constraint* (FCM_S) (Ahmed, dkk., 2002). Pada metode yang diusulkan, jarak piksel ke *mean* kluster (*centroid*) dan jarak piksel terhadap piksel tetangganya dihitung yang kemudian digunakan untuk memberikan bobot disimilaritas berdasarkan rata-rata keduanya. Walaupun telah menambahkan perhitungan ketetanggaan pada fungsi objektif, metode ini sensitif terhadap *outlier* dan membutuhkan waktu yang cukup lama. Oleh karena itu diusulkan *Fast Generalized FCM* (FGFCM) (Cai, dkk., 2007). Pada metode ini didefinisikan sebuah pengukuran disimilaritas yang berdasarkan ketetanggaan dan informasi spasial sebagai koefisien pembobot. Untuk mengurangi waktu eksekusi, FGFCM melakukan segmentasi gambar ditingkat abu-abu.

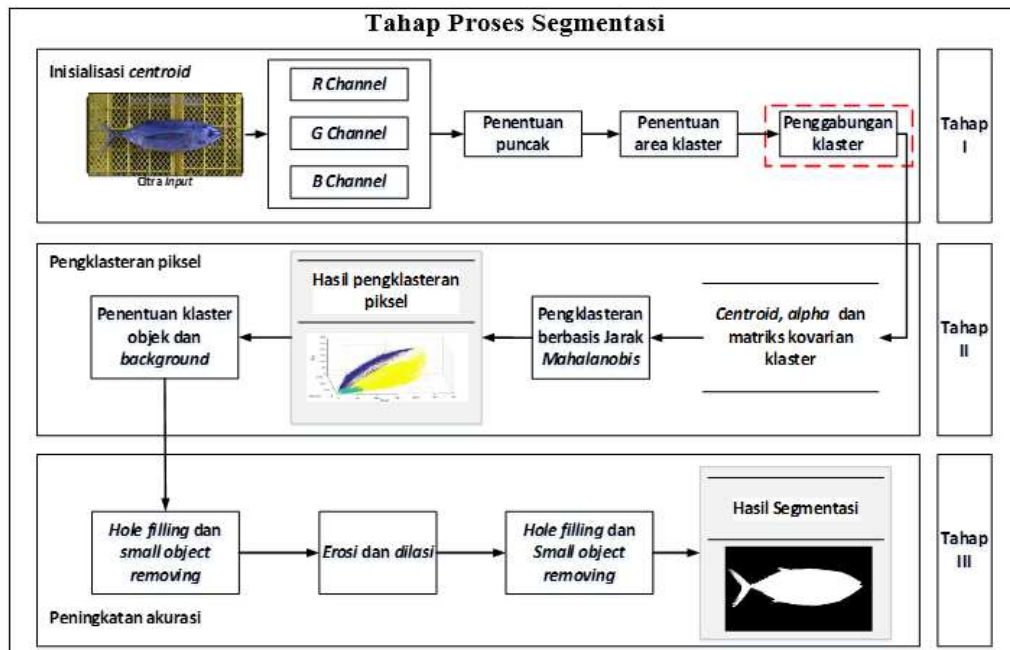
Beberapa peneliti mencoba meningkatkan performa dari FCM untuk segmentasi dengan melakukan penginisialisasian *centroid*. Diusulkan metode *Hybrid Ant Colony-Fuzzy C-Means* (AFHA) (Yu, dkk., 2010). Metode ini meningkatkan kualitas hasil pengklasteran dalam ruang fitur namun tidak efisien karena kompleksitas komputasinya. Untuk mengurangi kompleksitas komputasinya, maka diusulkan metode *Hybrid Histogram Thresholding* dan FCM (HTFCM), dimana *Histogram Thresholding* dimanfaatkan untuk mendapatkan inisialisasi *centroid* awal (Tan & Isa, 2011). Metode ini dapat mereduksi kompleksitas komputasi sekaligus memberikan hasil pengklasteran yang akurat dibandingkan dengan metode konvensional.

Metode pengklasteran konvensional berbasis jarak *Euclidean* dapat digunakan untuk mensegmentasi citra. Namun, metode tersebut tidak efektif jika diterapkan terhadap citra yang pikselnya membentuk kluster *hyperellipsoid* pada ruang fitur karena jarak *Euclidean* hanya menggunakan informasi *mean* dari kluster (Zhao, dkk., 2015b). Di sisi lain, metode pengklasteran berbasis jarak *Mahalanobis* lebih efektif untuk mengklaster citra yang pikselnya membentuk kelompok *hyperellipsoid* pada ruang fitur. Dalam rangka meningkatkan performa FCM, maka diusulkan FCM berbasis jarak *Mahalanobis* dengan mendefinisikan matriks dari piksel dan *mean*-nya (Liu, dkk., 2009). Namun, ketika titik data berada pada posisi yang sama dengan *centroid* kluster, disimilaritas antara keduanya tidak bernilai nol yang berarti metode tersebut tidak dapat mendefinisikan jarak dengan baik. Oleh karena itu, diusulkan FCM berbasis *Mahalanobis Distance* (MFCM) serta penambahan aturan baru pada fungsi objektif yang merefleksikan *kovarian* dari kluster (Zhao, dkk., 2015b). Namun, inisialisasi *centroid* dan derajat keanggotaan secara *random* mengakibatkan proses pengklasteran piksel menjadi tidak efisien dalam hal iterasi dan waktu komputasi.

Pada penelitian ini, penulis mengusulkan metode baru untuk segmentasi citra ikan tuna dengan *Mahalanobis Histogram Thresholding* (M-HT) dan *Mahalanobis Fuzzy C-Means* (MFCM). Metode yang diusulkan terdiri atas tiga tahapan utama, yaitu: inisialisasi *centroid*, pengklasteran piksel dan peningkatan akurasi. Metode yang diusulkan dapat melakukan segmentasi terhadap citra ikan tuna dengan jumlah iterasi dan waktu komputasi proses segmentasi yang lebih rendah dibandingkan dengan metode konvensional. Hal ini menunjukkan bahwa metode yang diusulkan dapat meningkatkan efisiensi dari proses segmentasi. Selain itu, juga dapat memberikan hasil segmentasi yang lebih akurat dibandingkan metode konvensional.

2. Metode

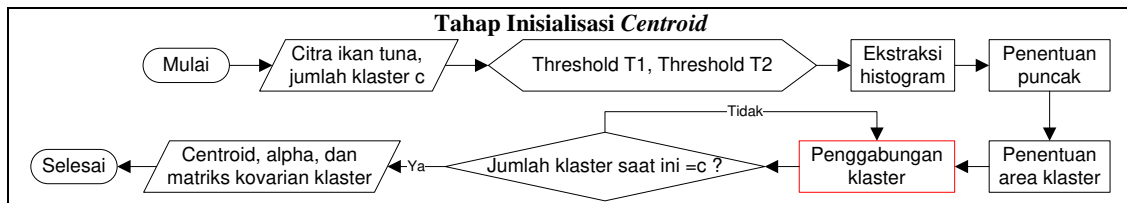
Ketiga tahapan metode yang diusulkan dan bagian kontribusi (kotak garis putus-putus) ditampilkan pada Gambar 1.



Gambar 1. Bagan Tahap Proses Segmentasi

2.1. Inisialisasi centroid

Input dari tahapan ini berupa citra original beberapa jenis ikan tuna, jumlah kluster c dan inisialisasi *threshold* T_1 dan T_2 . *Flowchart* tahap ini dapat dilihat pada Gambar 2.



Gambar 2. Flowchart Tahap Inisialisasi Centroid

Sub tahapan pertama dilakukan proses ekstraksi *histogram*. *Histogram* yang diperoleh akan dijadikan *input* pada tahapan berikutnya berdasarkan persamaan (1) sampai persamaan (3), dimana $0 \leq i \leq L-1$. Kedua, dilakukan proses penentuan puncak untuk setiap *histogram*. *Input* dari tahapan ini adalah *histogram* yang diperoleh pada subtahapan sebelumnya. *Output* berupa jumlah puncak pada tiap *histogram* dan intensitasnya. Langkah-langkah untuk menemukan puncak tersebut yaitu: (1) Bentuk *histogram* yang baru berdasarkan *histogram* yang diperoleh dari citra original dengan menggunakan persamaan (4), dimana s adalah *histogram* r , g atau b dan memenuhi $5 \leq i \leq L-6$. (2) Identifikasi seluruh puncak untuk tiap *histogram* dengan menggunakan persamaan (5), dimana $1 \leq i \leq L-2$. (3) Identifikasi semua lembah untuk tiap *histogram* dengan menggunakan persamaan (6), dimana $1 \leq i \leq L-2$. (4) Hilangkan semua lembah dan puncak berdasarkan *fuzzy rule base* yang ditunjukkan pada persamaan (7), dimana $1 \leq i \leq L-2$. (5) Identifikasi puncak pada tiap *histogram* dengan menguji titik yang terbentuk yang memiliki perubahan gradien positif atau negatif dan jumlah piksel yang lebih besar dari nilai *threshold* T_1 yang telah didefinisikan sebelumnya.

$$r(i) = x_i, \quad (1)$$

$$g(i) = x_i, \quad (2)$$

$$b(i) = x_i, \quad (3)$$

$$T_s(i) = \frac{(s(i-5) + s(i-4) + \dots + s(i) + \dots + s(i+4) + s(i+5))}{11}, \quad (4)$$

$$P_s = ((i, T_s(i)) | T_s(i) > T_s(i-1) \text{ dan } T_s(i) > T_s(i+1)), \quad (5)$$

$$V_s = ((i, T_s(i)) | T_s(i) < T_s(i-1) \text{ dan } T_s(i) < T_s(i+1)), \quad (6)$$

$$\begin{aligned} & \text{IF}(i \text{ is peak}) \text{AND } (T_{-s}(i+1) > T_{-s}(i-1)) \text{ THEN } (T_{-s}(i) = T_{-s}(i+1)) \\ & \text{IF}(i \text{ is peak}) \text{AND } (T_{-s}(i+1) < T_{-s}(i-1)) \text{ THEN } (T_{-s}(i) = T_{-s}(i-1)) \\ & \text{IF}(i \text{ is valley}) \text{AND } (T_{-s}(i+1) > T_{-s}(i-1)) \text{ THEN } (T_{-s}(i) = T_{-s}(i-1)) \\ & \text{IF}(i \text{ is valley}) \text{AND } (T_{-s}(i+1) < T_{-s}(i-1)) \text{ THEN } (T_{-s}(i) = T_{-s}(i+1)) \end{aligned} \quad (7)$$

Ketiga, proses penentuan area kluster. Tujuan dari tahapan ini adalah untuk menentukan area awal kluster berdasarkan jumlah *centroid* yang ada. *Input* dari tahapan ini adalah jumlah puncak yang diperoleh dari masing-masing *histogram*. *Output* dari tahapan ini berupa inisialisasi *centroid* dan area awalnya. Langkah-langkah untuk menentukan inisialisasi area kluster adalah sebagai berikut: (1) Mendapatkan semua kemungkinan *centroid* awal. Jumlah *centroid* diperoleh berdasarkan jumlah puncak x , y , dan z yang teridentifikasi. (2) Masukkan tiap titik data (piksel) ke dalam keanggotaan *centroid* terdekat dan berikan label sesuai dengan *centroid*-nya. (3) Eliminasi semua kluster *centroid* yang memiliki jumlah piksel dibawah *threshold* T_2 . (4) Kelompokkan ulang tiap piksel x_i ke dalam keanggotaan *centroid* μ_j terdekat. (5) *Update* tiap kluster *centroid* μ_j berdasarkan keanggotaan yang baru.

Keempat, proses penggabungan kluster. Proses penggabungan kluster memperhitungkan posisi *centroid* dan matriks *kovarian* kluster (Kelly, 1994). Langkah-langkah untuk penggabungan kluster adalah sebagai berikut: (1) Set jumlah kluster maksimal c . (2) Hitung jarak antar kluster menggunakan persamaan (8), dimana β diperoleh menggunakan persamaan (9) dan k_1 dan k_2 diperoleh menggunakan persamaan (10) dan persamaan (11), dimana i dan j adalah dua buah kluster yang diukur jaraknya, μ adalah *centroid* dari sebuah kluster, Σ^{-1} adalah invers dari *kovarian* masing-masing kluster, dan T adalah *transpose*. (3) Temukan dua buah kluster yang memiliki jarak terdekat. (4) *Update* kluster baru dengan menghitung anggota N_{new} , *centroid* μ_{new} dan *kovarian* Σ_{new} yang baru menggunakan persamaan (12) sampai dengan persamaan (14). (5) Kurangi jumlah *centroid*, ulangi langkah 2 sampai 5 hingga jumlah kluster maksimal c telah terpenuhi. (6) Hitung nilai α_j menggunakan persamaan (15), dimana jarak d_{ij} merupakan jarak antara piksel x_i terhadap *centroid* μ_j , dan Σ_j^{-1} merupakan invers dari matriks *kovarian* kluster ke- j yang dihitung menggunakan persamaan (16). (7) Berdasarkan kluster akhir yang terbentuk, dapatkan *centroid*, *alpha*, dan matriks *kovarian* masing-masing kluster.

$$r_{ij} = k_1 \beta^2, \quad (8)$$

$$\beta = \frac{\sqrt{k_1 k_2} - k_2}{k_1 - k_2}, \quad (9)$$

$$k_1 = [\mu_i^T \sum_j^{-1} \mu_i + \mu_j^T \sum_j^{-1} \mu_j - 2\mu_i^T \sum_j^{-1} \mu_j], \quad (10)$$

$$k_2 = [\mu_i^T \sum_i^{-1} \mu_i + \mu_j^T \sum_i^{-1} \mu_j - 2\mu_i^T \sum_i^{-1} \mu_j], \quad (11)$$

$$N_{new} = N_i + N_j, \quad (12)$$

$$\mu_{new} = \frac{N_i}{N_{new}} \mu_i + \frac{N_j}{N_{new}} \mu_j, \quad (13)$$

$$\Sigma_{new} = \frac{N_i - 1}{N_{new} - 1} \sum_i + \frac{N_j - 1}{N_{new} - 1} \sum_j + \frac{N_i N_j}{N_{new}(N_{new} - 1)} [(\mu_i - \mu_j)(\mu_i - \mu_j)^T]. \quad (14)$$

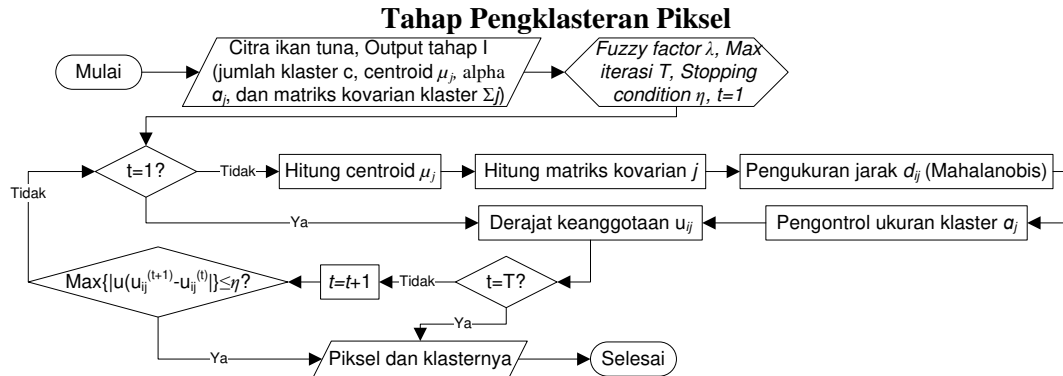
$$\alpha_j = \frac{\sum_{i=1}^N d_{ij}}{N}, \quad (15)$$

$$d_{ij} = (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j). \quad (16)$$

2.2. Pengklasteran piksel

Input dari tahapan ini adalah informasi piksel citra ikan tuna, matriks *kovarian* klaster, inisialisasi *centroid* dan *alpha*. *Output* yang diperoleh dari tahapan ini adalah piksel yang telah dikelompokkan ke dalam *c* klaster. *Flowchart* tahapan ini dapat dilihat pada Gambar 3.

Adapun penjelasan tiap langkah dari tahapan ini, yaitu: (1) *Input* informasi piksel citra ikan tuna yang digunakan pada tahap 1, inisialisasi *centroid*, matriks *kovarian* klaster, dan *alpha* yang diperoleh dari tahap 1. (2) Inisialisasi *fuzzy factor* λ , maksimum iterasi *T*, *stopping condition* η dan *iteration counter* $t=1$. (3) Jika $t \leq 1$, menuju pada langkah ke-8; Selain itu, lanjut pada langkah ke-4. (4) Mencari nilai *centroid* μ_j menggunakan persamaan (17), dimana *N* adalah banyaknya data, u_{ij} adalah derajat keanggotaan piksel ke-*i* terhadap klaster ke-*j* dan x_i adalah vektor fitur piksel ke-*i*. (5) Mencari matriks *kovarian* Σ_j untuk tiap klaster ke-*j* menggunakan persamaan (18), dimana *T* adalah *transpose*. Nilai *centroid* dan matriks *kovarian* untuk tiap klaster akan dijadikan acuan untuk mengetahui jarak tiap data terhadap *centroid*. (6) Hitung disimilaritas tiap piksel terhadap tiap *centroid* untuk mengetahui seberapa dekat jarak data piksel x_i terhadap *centroid* klaster μ_j dengan menggunakan jarak *Mahalanobis* d_{ij} seperti ditunjukkan pada persamaan (16). (7) Hitung nilai *alpha* α_j , dimana α_j sebagai variabel yang mengontrol ukuran klaster. α_j dihitung dengan menggunakan persamaan (19). (8) Hitung derajat keanggotaan u_{ij} untuk tiap piksel terhadap tiap klaster menggunakan persamaan (20), dimana d_{ij} adalah jarak piksel terhadap *centroid* klaster berdasarkan informasi dari ruang fitur, λ adalah *Fuzzy factor*, $|\Sigma_j|$ adalah determinan dari matriks *kovarian* klaster ke-*j*. (9) Jika $\max\{|u_{ij}^{(t+1)} - u_{ij}^{(t)}|\} < \eta$ atau penghitung iterasi $t=T$, kemudian berhenti; selain itu, $t=t+1$ dan melangkah kembali ke langkah ke-3. (10) Tentukan keanggotaan tiap klaster, kemudian tentukan klaster yang mewakili objek atau *background*. Jika jumlah klaster $c > 2$, gabung klaster *background* sehingga diperoleh dua klaster yang masing-masing mewakili area *background* dan objek.



Gambar 3. Flowchart Tahap Pengklasteran Piksel

$$\mu_j = \frac{\sum_{i=1}^N u_{ij} x_i}{\sum_{i=1}^N u_{ij}} \quad (17)$$

$$\Sigma_j = \frac{\sum_{i=1}^N u_{ij} (x_i - \mu_j)(x_i - \mu_j)^T}{\lambda \sum_{i=1}^N u_{ij}} \quad (18)$$

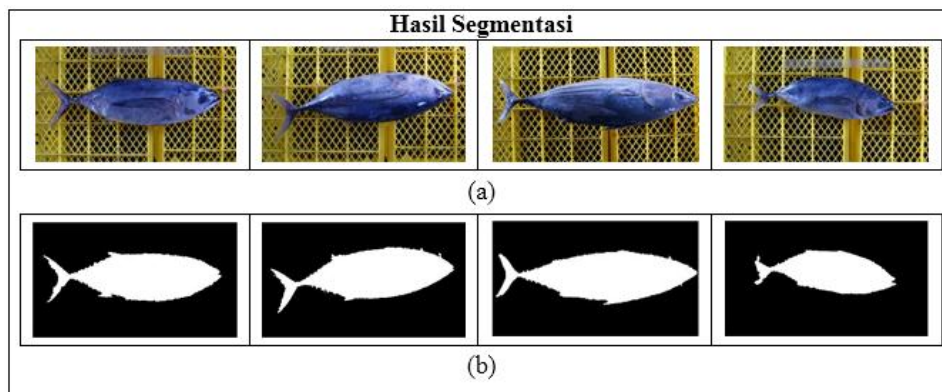
$$\alpha_j = \frac{\sum_{i=1}^N u_{ij}}{N} \quad (19)$$

$$u_{ij} = \frac{\alpha_j \exp(-\frac{d_{ij} + \lambda \log|\sum j|}{\lambda})}{\sum_{j=1}^c \alpha_j \exp(-\frac{d_{ij} + \lambda \log|\sum j|}{\lambda})} \quad (20)$$

2.3. Peningkatan akurasi

Tahapan ini bertujuan untuk meningkatkan akurasi hasil pengklasteran piksel. *Input* tahapan ini berupa hasil pengklasteran tahap sebelumnya. *Output* yang akan diperoleh adalah hasil segmentasi yang akurat. Hasil pengklasteran piksel yang diperoleh biasanya menyisakan beberapa piksel yang misklasifikasi jika dilihat dalam ruang spasial. *Small object removing* diterapkan terhadap sekumpulan piksel *background* yang teridentifikasi sebagai objek yang luasnya dibawah 3.000 piksel. Kemudian *Hole filling* untuk menutupi bagian tengah objek yang dianggap sebagai *background*.

Selain itu, hasil pengklasteran piksel jika dipetakan dalam ruang spasial menunjukkan terdapat piksel pada bagian tepi objek yang misklasifikasi. Sehingga, perlu untuk direduksi sesuai dengan bagian objek yang sesungguhnya dengan melakukan operasi erosi dengan *structuring element* SE_1 disc ukuran 3x3 dan dilasi dengan SE_2 disc ukuran 5x5. Kemudian, subtahapan terakhir ini kembali dilakukan *hole filling* dan *small object removing*. Setelah tahap ini selesai maka akan diperoleh hasil segmentasi citra ikan tuna seperti ditunjukkan pada Gambar 4.



Gambar 4. Hasil Segmentasi Sampel (a) Citra Original dan (b) Citra Hasil Segmentasi

3. Hasil dan pembahasan

Pada penelitian ini digunakan 30 citra ikan tuna, dimana terdapat 10 citra untuk tiap jenis ikan tuna *Skipjack*, *Yellowfin* dan *Bigeye*. Citra dalam ruang warna RGB dan berukuran 217x387 piksel. Pada metode yang diusulkan ditetapkan nilai T_1 dan T_2 sebesar 300 dan 700 berturut-turut. Setelah itu dilakukan penentuan jumlah klaster c yang optimal berdasarkan dataset yang digunakan. Berdasarkan hasil percobaan yang dilakukan terhadap seluruh citra, maka diperoleh jumlah klaster yang paling optimal adalah 3. Sedangkan untuk nilai *fuzzy factor* dan *stopping condition* η diset sebesar 1,5 dan 0,00001.

Adapun evaluasi yang kami lakukan adalah menghitung jumlah iterasi proses segmentasi, waktu segmentasi, *Missclassification Error* (ME) dan Akurasi. ME dan Akurasi dihitung menggunakan persamaan (21) dan persamaan (22), dimana B_o adalah *background groundtruth* dan F_o adalah *foreground groundtruth*, sedangkan B_T dan F_T dinotasikan sebagai *background* dan *foreground* dari citra hasil segmentasi yang diperoleh, dimana mc adalah jumlah piksel yang misklasifikasi dan tp adalah total piksel yang seharusnya terklasifikasi dengan benar.

$$ME = 1 - \frac{|B_o \cap B_T| + |F_o \cap F_T|}{|B_o| + |F_o|}, \quad (21)$$

$$AS(\%) = \left[1 - \left(\frac{mc}{tp} \right) \right] \times 100\%, \quad (22)$$

Untuk menunjukkan keunggulan dari metode yang diusulkan (M-HT MFCM), maka kami membandingkan hasil evaluasi dari metode yang kami usulkan dengan metode konvensional (MFCM). Evaluasi hasil ujicoba dataset yang digunakan berdasarkan jenis ikan tuna dapat dilihat pada Tabel 1. Metode yang diusulkan (P) mampu memberikan rata-rata iterasi, waktu komputasi dan ME yang lebih rendah, serta akurasi yang lebih tinggi dibandingkan dengan metode konvensional (C) untuk semua jenis ikan tuna. Hal ini menunjukkan bahwa metode yang diusulkan dapat meningkatkan efisiensi metode segmentasi sekaligus memberikan hasil segmentasi yang akurat.

Tabel 1. Perbandingan Performa Metode Usulan dan Metode Konvensional

No.	Jenis Ikan Tuna	ME (Persen)		Akurasi (Persen)		Iterasi		Waktu Segmentasi (Detik)	
		P	C	P	C	P	C	P	C
1.	<i>Skipjack</i>	1.26	6.97	98.74	93.03	47	81	118.10	197.07
2.	<i>Yellowfin</i>	1.10	16.91	98.90	83.09	72	98	180.18	250.97
3.	<i>Bigeye</i>	2.02	8.06	97.98	91.94	79	117	198.87	295.49

Berdasarkan hasil yang diperoleh dari metode yang diusulkan, dapat diketahui bahwa secara umum hasil yang diperoleh telah mencapai tujuan dari penelitian ini. Pada tahap inisialisasi *centroid* menggunakan *Mahalanobis Histogram Thresholding*, metode penggabungan kluster dengan memanfaatkan pengukuran jarak berbasis jarak *Mahalanobis* yang tidak hanya memerhatikan posisi kluster namun juga *kovarian* kluster secara umum dapat menghasilkan hasil penggabungan kluster yang tepat. Sehingga, *centroid* berikut matriks *kovarian* dan *alpha* yang diperoleh tersebut dapat mengoptimalkan derajat keanggotaan pada iterasi pertama proses pengklasteran piksel, sehingga mereduksi jumlah iterasi untuk mencapai titik konvergen. Reduksi iterasi tersebut mengakibatkan metode yang diusulkan menjadi lebih efisien dalam hal iterasi dan waktu komputasi.

Metode yang diusulkan dapat menghasilkan ME yang rendah dan akurasi yang tinggi karena proses pengklasteran tidak hanya memerhatikan posisi *centroid* melainkan juga memperhitungkan *kovarian* dari kluster. Sehingga, objek bagian ikan tuna dari yang berwarna biru gelap hingga biru terang dapat dimasukkan ke dalam satu kluster yang sama. ME yang rendah dan akurasi yang tinggi juga disebabkan karena pada saat proses pengklasteran piksel, jumlah kluster ditentukan sebanyak tiga kluster. Berdasarkan hasil percobaan, jumlah kluster $c=3$ memberikan hasil ME yang paling rendah. Jika dilakukan analisa lebih mendalam terhadap citra ikan tuna, pembagian wilayah ikan tuna dalam ruang fitur ke dalam tiga kluster dapat memberikan hasil yang paling optimal. Hal ini disebabkan karena: 1) Bagian ikan tuna memiliki warna yang bervariasi dari biru gelap hingga biru terang, 2) *Background* ikan tuna terdiri atas bagian yang gelap (hitam) dan bagian berwarna terang (kuning). Jumlah piksel bagian citra ikan tuna, *background* hitam dan *background* berwarna kuning hampir memiliki jumlah yang sama banyaknya sehingga ketika dilakukan pengklasteran maka bagian ikan tuna, *background* hitam dan *background* kuning masing-masing akan menjadi satu kluster.

Pada tahap proses peningkatan akurasi hasil pengklasteran piksel, diperoleh hasil pengklasteran piksel yang lebih baik. Hal ini disebabkan objek-objek lain selain ikan tuna diubah menjadi kluster *background*. Sehingga dengan demikian objek yang tidak bergabung dengan bagian tubuh ikan tuna akan menjadi anggota kluster *background*, begitupun sebaliknya.

4. Kesimpulan

Berdasarkan hasil uji coba dan analisis hasil pengujian pada metode segmentasi citra ikan tuna dengan *Mahalanobis Histogram Thresholding* (M-HT) dan *Mahalanobis Fuzzy C-*

Means (MFCM), dapat disimpulkan bahwa metode yang diusulkan dapat meningkatkan efisiensi proses segmentasi dalam hal iterasi dan waktu segmentasi. Selain itu, metode yang diusulkan juga dapat memberikan hasil segmentasi yang akurat dibandingkan dengan metode konvensional. Rata-rata jumlah iterasi pada proses segmentasi sebanyak 66 iterasi, 31 iterasi lebih sedikit dibanding metode konvensional. Rata-rata waktu segmentasi untuk seluruh sampel sebesar 162,03 detik, 84,40 detik lebih cepat dibanding metode konvensional. Akurasi untuk seluruh sampel menghasilkan rata-rata akurasi segmentasi 98,54% dengan tingkat *Missclassification Error* 1,46%, 17,24% lebih akurat dibandingkan dengan metode konvensional. Pengembangan penelitian lebih lanjut melakukan proses pengklasteran dengan menggabungkan informasi dari ruang fitur dan ruang spasial agar dapat diperoleh hasil segmentasi yang lebih akurat.

Referensi

- Ahmed, M. N., Yamany, S. M., Mohamed, N., Farag, A. A., & Moriarty, T. 2002. A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *Medical Imaging, IEEE Transactions on*, 21(3), 193-199.
- Arifin, A. Z., & Asano, A. 2006. Image segmentation by histogram thresholding using hierarchical cluster analysis. *Pattern Recognition Letters*, 27(13), 1515-1521.
- Bezdek, J. C., Ehrlich, R., & Full, W. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2), 191-203.
- Cai, W., Chen, S., & Zhang, D. 2007. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition*, 40(3), 825-838.
- Ghosh, M., Das, D., Chakraborty, C., & Ray, A. K. 2010. Automated leukocyte recognition using fuzzy divergence. *Micron*, 41(7), 840-846.
- Jati, A., Singh, G., Mukherjee, R., Ghosh, M., Konar, A., Chakraborty, C., & Nagar, A. K. 2014. Automatic leukocyte nucleus segmentation by intuitionistic fuzzy divergence based thresholding. *Micron*, 58, 55-65.
- Kelly, P. M. 1994. An algorithm for merging hyperellipsoidal clusters. *Los Alamos National Laboratory, Tech. Rep.*
- Liu, H. C., Jeng, B. C., Yih, J. M., & Yu, Y. K. 2009. Fuzzy C-means algorithm based on standard mahalanobis distances. In *Proceedings of the 2009 International Symposium on Information Processing* (pp. 422-427).
- Liu, X., Lin, L., & Yuille, A. 2013. Robust region grouping via internal patch statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1931-1938).
- Puspita, S.D., Arifin, A.Z. & Khotimah, N., 2015. Penggunaan multi texton co-occurrence descriptor untuk klasifikasi ikan tuna. *Jurnal Teknik ITS*, 4(1),1–6.
- Tan, K. S., & Isa, N. A. M. 2011. Color image segmentation using histogram thresholding–Fuzzy C-means hybrid approach. *Pattern Recognition*, 44(1), 1-15.
- Yao, H., Duan, Q., Li, D., & Wang, J. 2013. An improved K-means clustering algorithm for fish image segmentation. *Mathematical and Computer Modelling*, 58(3), 790-798.
- Yu, Z., Au, O. C., Zou, R., Yu, W., & Tian, J. 2010. An adaptive unsupervised approach toward pixel clustering and color image segmentation. *Pattern Recognition*, 43(5), 1889-1906.
- Zhao, F., Liu, H., & Fan, J. 2015a. A multiobjective spatial fuzzy clustering algorithm for image segmentation. *Applied Soft Computing*, 30, 48-57.
- Zhao, X., Li, Y., & Zhao, Q. 2015b. Mahalanobis distance based on fuzzy clustering algorithm for image segmentation. *Digital Signal Processing*, 43, 8-16.